A Fuzzy Based Priority Approach in Mobile Sensor Network Coverage

Bahareh J. Farahani, Hossein Ghaffarian ,Mahmood Fathy
Computer Engineering School, Iran University of Science and Technology, Tehran, Iran
Farahani.ict@gmail.com, Ghaffarian@iust.ac.ir, MahFathy@iust.ac.ir

Abstract— **In this paper a new fuzzy based approach for improving network coverage in wireless mobile sensor networks is proposed. In the proposed approach firstly each mobile sensor node determines its neighbors and its distance from borders and obstacles. According to these values, fuzzy inference engine calculates the priority of node for movement. Then according to the priority, in turn, nodes move away from each other to increase coverage area in the target field. Simulation results show that our fuzzy approach can reach higher degree of coverage against other common approaches like FOA, VEC and TRI algorithms.**

Index Terms— **Mobile Sensor Network, Coverage, Fuzzy Priority, Movement**

I. Introduction

Advances in Micro Electro Mechanical Systems (MEMS), embedded processors and wireless communications have enabled the expansion of multi-functional, low-cost and tiny sensor nodes which can sense the environment, perform data processing and communicate with each other over short distances. Wireless sensor networks (WSNs) are playing an increasingly significant role in a wide range of applications such as environmental conditions monitoring, wildlife monitoring, security surveillance in military and battle-fields, nuclear, smart homes and offices, improved medical treatment, industrial diagnosis, etc. The concept of area coverage can be considered as a measure of the quality of service (QoS) in a wireless sensor network. Random deployment of nodes often does not guarantee full coverage, resulting in accumulation of nodes at certain parts of the sensing field while leaving other parts deprived of nodes. Some of the deployment strategies take advantage of mobility to relocate nodes to sparsely covered regions after an initial random deployment to improve coverage. Thus as a solution, mobile sensor networks (MSNs) can be used for network coverage improvements.

In this paper, a novel algorithm for placement of mobile sensors is designed. The main result of this paper is as follows:

A Fuzzy logic method is described to find the movement priority for mobile sensors and a distributed fault tolerant, self deployment and iterative algorithm to cover an area. Mobile sensors only need to have knowledge of their location and with the assistance of local communication they will act in a completely distributed fashion. To demonstrate the generality and effectiveness of this algorithm, our algorithm is compared with FOA [1], TRI [2] and VEC [3] algorithms. Comparison will show that our algorithm exhibit better performance.

This paper is organized as follows: In section II, an overview on related approaches is presented. Section III consists of problem setup and an overview and analysis of effective parameters on the position of mobile sensors. How to fuzzy reasoning and proposed method have been describe in section IV and the results of performance evaluation of our method, FOA, VEC and TRI algorithms have been presented in section V. Finally the paper has been concluded in section VI.

II. Related Approaches

Dealing with coverage problem, researchers have proposed numerous solutions. Some algorithms are based on the notion of potential field [4], [5], [6] and virtual forces[7],[8] after an initial random deployment.

The next three distributed self deployment algorithms proposed in [3] known as the VEC, VOR, and Minimax are based on the structure of Voronoi diagram in which nodes are relocated to fill up coverage holes.

The vector-based algorithm, VEC, pushes nodes away from densely covered areas to sparsely covered areas. Two nodes exert a repulsive force when they are too close to each other. VOR is a greedy strategy that pulls nodes towards the locations of their local maximum coverage holes. The MiniMax algorithm is very similar to VOR; it moves a node inside its Voronoi polygon, such that, the distance from its farthest Voronoi vertex is minimized. In [9] and [10], an incremental and greedy self-deployment algorithm is presented for mobile sensor networks in which nodes are deployed one at a time into an unknown environment. Each node makes use of the information gathered by previously deployed nodes to determine its optimal deployment location. Conceptually, it is similar to the frontier-based approach [11]; however, in this case, occupancy maps are built from live sensory data and are analyzed to find frontiers between the free space and the unknown space.

The TRI algorithm is proposed in [2]. The basic idea of TRI algorithm is to adjust the distance between two Delaunay neighbors to $\sqrt{3}R_{sen}$. After several rounds of such adjustments, the layout of the network will be

DOI: 01.ijns.01.02.09

close to the ideal equilateral triangle layout. As a result, the coverage area of the network will be maximized.

The ATRI algorithm [2] is offered in order to reduce the back-and-forth movement of nodes.

The algorithms described in previous apply to networks where all the nodes are capable of moving around. However, there is a high cost associated to make each node mobile; a balance can be achieved by using a combination of static and mobile nodes, usually referred to as hybrid sensor networks, while still ensuring sufficient coverage. In [12], such a protocol is described, called the bidding protocol, where the problem is reduced to the well-known NP-hard set-covering problem.In [1] a Fuzzy Optimization Algorithm (FOA) is proposed to deploy sensors efficiently. Authors use the number of neighbors and average Euclidean distance between sensors as inputs of the proposed Fuzzy reasoning engine algorithm and get a distance measure as output. In proposed algorithm each sensor calculates the forces imposed by the other sensors within its communication range. Then by applying Coulomb's law and accumulating the vectors of all the forces, the sensor can calculate the magnitude and the angle of its next movement vector in order to move toward the new next location.

III. Problem Setup and Effective Components

Before describing proposed method, it is necessary to describe some related concepts and effective components on the position of a mobile sensor in a target field. In proposed method, effects of these components have been collected.

A. Preliminaries

Two-dimensional deployment: It is assumed that all devices are located in two dimensions plan.
Location awareness: Knowing the location of sensor nodes is important for our algorithm. Many techniques have been proposed to supply this information for mobile sensors [13],[14].
Boundary and Obstacle detecting: It is assumed that each sensor node is equipped with an ultrasonic obstacle detecting module which makes it possible to detect boundaries when it moves close enough to it.
Sensing model: Each sensor node in our algorithm is associated with a sensing area which is represented by a circle with the same Sensing Radius. This Circular area is called Sensing Range and depends on several factors in the network [15]. A sensor node can detect all the events in its Sensing Range. The areas which are not covered by these circles are called coverage holes.
Radio range: Apart from Sensing Range, each sensor has a Communication Range which is much longer than Sensing Range. When a node broadcasts a message, all the sensor nodes in its Communication Range will receive this message.
Sensor equipment: Each mobile sensor is equipped with a battery that provides sufficient power to move and exchange information with other sensors. There is also a memory in each sensor which used to preserve require information of network elements.

B. Effect of Adjacent Nodes

Cellular networks are a good example of an engineering task for supporting and presenting good quality of coverage in network. Due to easier calculations, in these networks, engineers have used hexagonal shapes for each cell. It has been proven [2] that in ideal layout, the sensors are located in vertices of equilateral Delaunay triangles [16] with edge length of $\sqrt{3}R_{ser}$. By this layout, the coverage area of nodes will be maximized with the minimum coverage overlap.

Using this value as a standard distance between neighbor nodes, if the distance between two nodes becomes shorter than this distance, the virtual force between them will push them to move away from each other.

C. Effect of Borders and Obstacles

Coverage in the presence of obstacles is a challenging problem. In order to make the deployment algorithm more practical, the sensors must be able to avoid obstacles and boundaries. Because an accurate map of the sensing area may not always be available before the deployment, it is assumed that each sensor is equipped with an ultrasonic obstacle-detecting module, which makes it possible to detect obstacles when it moves close enough to the obstacles. In order to avoid the coverage gap or overlap between nodes and boundaries, intuitively, the optimum distance between adjacent nodes and boundaries should be shorter than $\sqrt{3}R_{ser}$.

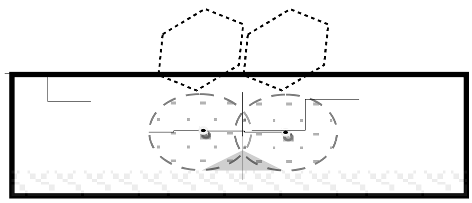


Figure 1. The effect of adjacent nodes causes a Figure 1. distance between them.

In Fig. 1, if the dashed line between two cells is considered as a boundary, the distance between the node and the boundary should be set $\sqrt{3}/2R_{ser}$, but using this approach will make some uncovered triangular holes beside the boundary. In order to avoid the coverage gap, as shown in Fig. 2, the point P must touch the boundary. By preliminary mathematics in geometry, an effective distance between boundaries or obstacles and mobile sensor nodes can be calculated. As mentioned in section *B*, the distance of two adjacent mobile sensor nodes has to be adjusted to $\sqrt{3}R_{ser}$. Thus, the distance between a node and a boundary must

DOI: 01.ijns.01.02.09

to be adjusted to R_{sense} instead of $(\sqrt{3}/2R_{sens}$as shown in below calculation.

$$EffectiveDist = \sqrt{R_{sense}{}^2 - \left(\frac{\sqrt{3}}{2}R_{sense}\right)^2} = \frac{R_{sense}}{2} \text{-1}$$

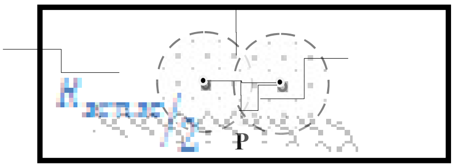


Figure 2. The effect of a border or obstacle on a node causes a Figure 1. distance between them.

D. Effect of Corners

A corner is the junction of at least two different boundaries. Thereby the proposed effective distance that has been calculated in section *C*, is not efficient here. For achieving a full coverage at corners, sensor must become close enough to the furthest point at the corner. Therefore mobile sensor node must stand at distance R_{ser} from the furthest point of corner. This value does not affected by different angels of corners. In the case of multiple corners, sensor will move toward the farthest vertex of corners and stop when the farthest corner vertex can be seen within the sensing radius of mobile node.

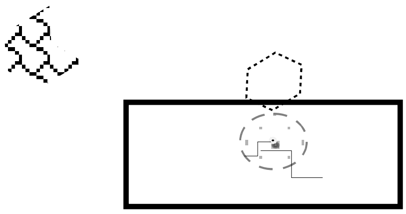


Figure 3. The effect of a corner on a node causes a Figure distance from farthest vertex of corner.

IV. Proposed Fuzzy based Approach

Fuzzy reasoning tries to continue the way of natural reasoning of human. It can present more smooth results than common proposed methods in reasoning. How to use fuzzy reasoning in a manner to make decision procedures for deployment of mobile sensor nodes in a target field is described in this section.

A. Fuzzy Based Priority Coverage Algorithm (FBPC)

Proposed method consists of two phases: First gathering information of neighbors and determining the priority of mobile sensor node to move and second calculating movement vector. At the first phase, the nodes try to find other mobile sensor nodes within their communication range and also detect boundaries and obstacles and the distance from them.

After being activated, each node waits for a random time and then broadcasts a HELLO message including its position in the target area. The sensors which are in the communication range of that node will get this message and will wait for a specific time (T_{wait}) to receive all possible HELLO messages from the network. In this interval time, nodes will start to update their *Neighboring Matrix* that is consists of gathered information from other mobile nodes around it. The mobile nodes which have some changes in their *neighboring matrix* will broadcast their information as a new HELLO message and will set a counter T_{thresh} ($T_{threshold} \ll T_w$). If within T_{thresh} no new HELLO message is received, the node starts to use fuzzy controller to define its priority for movement.

The proposed fuzzy controller has two inputs:

- Number of mobile nodes in the sensing range of the node
- Length of *Sensing Shadow* on boundaries or obstacles.

Definition 1. In a two-dimensional mode, *sensing shadow* is the length of boundary which is inside the sensing range of the sensor node.

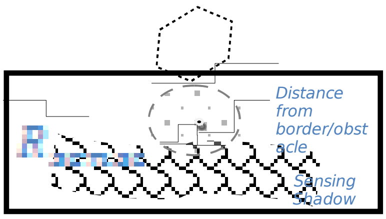


Figure 4. A schematic of sensing shadow.

Sensing shadow can be calculated in the same way of equation (1) and its value is between $[0 - 2R_{sen}$. If the distance between mobile node and boundary or obstacle is greater than or equal to R_{ser}, the sensing shadow will be zero and if mobile node sticks to the boundary or obstacle, the sensing shadow will be $2R_{ser}$. This parameter is used to define *the priority of standing in place* for mobile sensor nodes. The nodes with minimum distance from borders or obstacles have highest priority for standing and should move after their adjacent nodes.

The maximum number of hexagonal neighbor cells is six; therefore in the proposed algorithm it is assumed that more than six neighbors is the worst case for initial deployment. So the number of neighbors is limited to ten as an upper bound for our controller so the controller will omit other extra neighbors automatically. In order to normalizing the value of *sensing shadow* into [0-1], it is divided by $2R_{ser}$.

The linguistic variables to represent the number of mobile nodes in the sensing range of the node are divided into three levels: *high*, *average* and *low*; and those to represent the Length of *Sensing Shadow* are also divided into three levels: *high*, *average* and *low*. The consequent is divided into five levels: *very high*,

DOI: 01.ijns.01.02.09

high, a*verage, very low* and *low*. Table 1 summaries the rules and consequents.

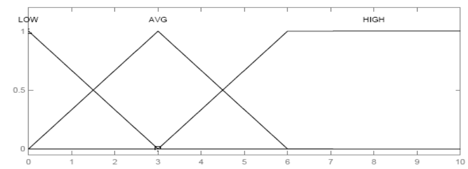
One example of rules is as follows:

IF the number of mobile nodes in the sensing range of sensor *i* is *high* and length of *Sensing Shadow* of sensor *i* on boundaries or obstacles is *low*, THEN the standing priority result of sensor *i* will be *very high*.

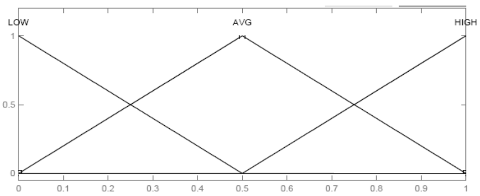
TABLE 1.
RULE BASE OF THE PROPOSED FUZZY STANDING PRIORITY CONTROLLER

Priority Controller Rules		***Neighbor Number***		
		Low	***Average***	***High***
Sensing Shadow	***Low***	*High*	*Very High*	*Very High*
	Average	*Low*	*Average*	*High*
	High	*Very Low*	*Low*	*High*

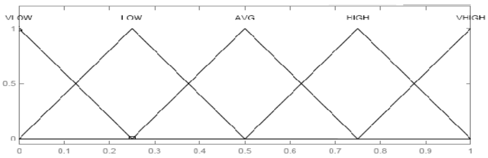
Fig. 5 shows input and output membership functions of our proposed fuzzy controller. The nodes with lowest standing priority must move first in a group of adjacent mobile sensor networks.



a. Neighbor Number membership function



b. Sensing shadow membership function



c. Priority membership function

Figure 5. Membership functions of the proposed Fuzzy Priority controller.

At this stage, there is no need for nodes to rebroadcast information of their priorities, because each mobile sensor can easily calculate the priority of its neighbors. At the second phase, each node with minimum standing priority can move before other nodes. So sensor will calculate its MOVEMENT VECTOR based on forces of its neighbor nodes, boundaries, obstacles and corners. In our model, each node s_i is subjected to three kinds of forces: (1) an attractive or repulsive force F_{ij}, by another node s_j depending on its distance and orientation from s_i, (2) a repulsive force F_{iB}, exerted by obstacles or boundaries, and (3) a repulsive force F_{iC}, exerted by corners. The net force on a sensor s_i is the vector sum of all the above forces. Table 2 summaries these three types of forces.

After calculating movement vector and defining destination, the node will send a GOODBYE packet that consists of the new calculated position of mobile node. Then lowest priority mobile node runs into the new position. Other nodes will update their neighboring matrix information after receiving GOODBYE packet and use this updated version in their calculation to find their final position.

After moving all nodes, this algorithm will be repeated again until nodes have not any changes in their positions. Algorithm 1 shows a pseudo code of the proposed Fuzzy Based Priority Coverage (FBPC) algorithm.

```
1:    FBPC ()
2:     {
3:       while ()
4:        {
5:         Stand for a random time t
6:         Broadcast a HELLO packet
7:            Stand for Twait to receive all possible HELLO
8:    packets
9:         Update Neighboring matrix
10:        If there are some changes in Neighboring matrix
11:          {
12:            Broadcast a HELLO packet.
13:            Stand for Tthreshold
14:              If within Tthreshold some new HELLO message
15:             is received
16:                  go to line 8
17:          }
18:            If there are not any changes in the position of
19:         node and its adjacent nodes
20:                  break;
21:        Calculate Sensing Shadow
22:         Determine movement turning according to the
23:         standing priority of node using Fuzzy priority
24:         controller
25:            Stand until receiving movement turning of the
26:         nodes and update the Neighboring matrix
27:         according to the
28:         information of the receiving GOODBYE packets
29:            Calculate MOVEMENT VECTOR according to
30:         the Neighboring matrix
31:            Broadcast GOODBYE packet and travel to the
32:         new position in target field
        }
     }
```


DOI: 01.ijns.01.02.09

TABLE 2.
TYPES OF FORCES EACH NODE ENCOUNTERS

Force	Related Standard Distance
F_{ij}=Standard Distance between nodes -d(s_i,s_j)	$\sqrt{3}R_{sense}$
F_{iB}=Standard Distance from boundary -d(s_i,boundary)	$R_{sense}/2$
F_{iC}=Standard Distance from corner-d(s_i,boundary)	R_{sense}

A Robust Algorithm

In some situations, a sensor may calculate a new position; but when it wants to move, because of some environmental conditions like holes, dead-end paths, etc; it could not reach destination target position. In such conditions FBPC algorithm is robust; because if one of the nodes cannot move to its final position, in the next round of algorithm, it can contribute in new making decision routine according to the information of its new position, not previous announced information. Therefore, FBPC is a fault tolerant algorithm that can tolerate probable mistakes and can correct its behavior in further rounds. Even if there are some errors in collecting local messages via network, FBPC can recover its action in consequent rounds.

V. PERFORMANCE EVALUATION

A. Objectives and Metrics

Our deployment protocols have been implemented in Matlab (version R2008a). Our objective in conducting this evaluation study is: by comparing FOA, TRI, VEC and our algorithm giving some insight on choosing suitable algorithm in different situations.

The simulations are run under different sensor densities, which determines the sensor coverage that can be reached and the difficulty to reach it. In a 100 m ×100 m target field, five different numbers of sensors have been distributed, ranging from 60 to 140 in increment of 20 sensors. The initial deployment of proposed algorithm, FOA and also VEC algorithm follow the random distribution, however in TRI, according to [2], at the beginning of the experiment, sensors are randomly placed within a 1m × 1m compact square, which is centered at point (50m, 50m). Then the nodes explode to a large evenly deployed layout. To evaluate each metric under different parameter setting, 5 experiments based on different initial distribution are run and the average results are beening calculated. The Sensing Range is set to be 6 meters and 20 meters as the Communication Range.

B. Simulation Results

In order to evaluate the performance and cost of the three algorithms, three metrics are measured for each simulation round: total coverage, average moving distance and average number of rounds. Fig. 6 shows the total coverage of three algorithms as the number of sensors increases.

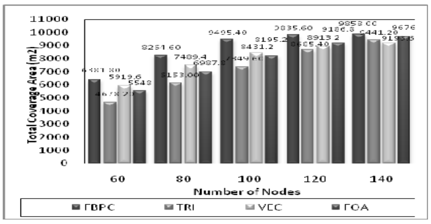


Figure 6. Total coverage

From the figure, it can be seen that the FBPC algorithm prepares the best total coverage. The primary reason is that unlike other algorithms, FBPC paysattention to obstacles and boundaries and keeps distance from them in order not to waste its sensing range.

Also, it can be seen that in sparse conditions, VEC algorithm stands in the second place, but in dense conditions, VEC is replaced by FOA.

Another dimension of evaluation is the average moving distance. As depicted in Fig.7, TRI algorithm performs the worst in average moving distance among FBPC, FOA, and VEC, Because in TRI mobile nodes are located in a unique point in target field and after calculating their accurate positions, they will move. Thus they must move more.

After TRI, FBPC algorithm has the bigger average moving distance. This happens because it tries to do a fast deployment and reach to a high level of coverage. Also in FBPC, nodes move away from borders and obstacles. Results of VEC and FOA can be compared with their coverage values. *More coverage needs more moving distance.*

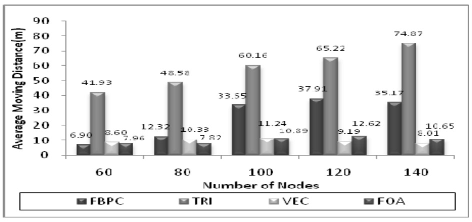


Figure 7. Average moving distance

The number of rounds in FOA algorithm is high. Although some of the authors disregard the effect of message complexity in increasing energy consumption, this cannot be ignored by increase in number of rounds. Calculated from Robomote [17],[3], approximately, to move a sensor one meter consumes a similar amount of energy as transmitting 300 messages. Assume that in each round each sensor transmits one packet for a network with 100 sensors, by running the FOA algorithm after each 3 rounds and 30 rounds, the movement distance of nodes will be increased 1m and

DOI: 01.ijns.01.02.09

10m, respectively. Moreover it is assumed that all packets can be transmitted without any error.

Random distribution of sensors occurs in urgent cases or for some special application. In such an urgent situation, the fast deployment of the sensors is in the first priority. In this case the performance of the FOA algorithm is not applicable at all, whereas FBPC algorithm presents better coverage with fast reaction.

Fig. 8 shows the result of average number of rounds in algorithms.

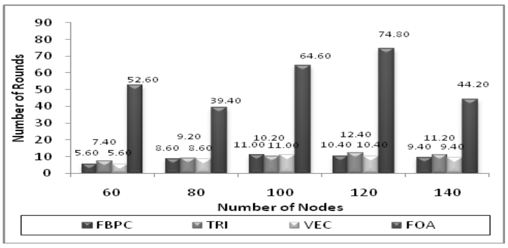


Figure 8. Average number of rounds

As it can be seen, FBPC and VEC achieve quite similar number of rounds and they can deploy very fast. As this figure shows, FOA algorithm has the worst performance.

Selecting the random intervals for waiting in FBPC algorithm in order to decrease the number of collision during the packet transformation, not only decreases the effect of not arriving a packet to the destination but also the next following packets can cover the effect of this error. While other existence algorithms act poorly and don't pay much attention in sending the messages and exchanging information.

VI. Conclusion and Feature Works

In this paper, a new fault tolerant deployment fuzzy based algorithm is proposed for unattended mobile sensor networks, called, the Fuzzy Based Priority Coverage (FBPC) algorithm. Algorithm is run in a completely distributed fashion by each node based only on local communications. Proposed method uses number of neighbors and distance from borders and obstacles as inputs of a fuzzy inference engine. The output is the actual priority of movement for sensor node. Nodes move away according to their priorities. Simulation results show that proposed method received a very good coverage comparing other common approaches, especially in the sparse networks.

For future works, we consider to improve the ability of our algorithm in sparse networks. Improving network coverage in sparse conditions will result in reducing overall cost of network.

References

[1] Hainin Shu, Qilian Liang, "Fuzzy optimization for distributed sensor deployment", *IEEE Wireless Communications and Networking Conference*, Vol. 3, pp.1903 – 1908, 13-17 March 2005.

[2] Ma, M. and Yang, Y., "Adaptive Triangular Deployment Algorithm for Unattended Mobile Sensor Networks", *IEEE Transactions on Computers*, Vol. 56, NO. 7, pp. 946 – 847, JULY 2007.

[3] Wang, G., Cao, G. and La Porta, T. ,"Movement-Assisted Sensor Deployment", *IEEE Transaction on Mobile Computing*, Vol.5, no.6, pp. 640–652, June 2006.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *International Journal of Robotics Research,* pp. 90–98, 1986.

[5] A. Howard, M. Mataric and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem", *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems*, Fukuoka, Japan, pp. 299–308,June 2002.

[6] S. Poduri, G.S. Sukhatme, "Constrained coverage in mobile sensor networks", *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, pp. 40–50, April–May 2004.

[7] Y. Zou, K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks", *Transactions on IEEE Embedded Computing Systems*, pp. 61–91, 2004.

[8] Y. Zou, K. Chakrabarty, "Sensor deployment and target localization based on virtual forces", *Proceedings of IEEE Infocom*, INFOCOM'03, San Francisco, CA, pp. 1293–1303, April 2003.

[9] A. Howard, M. Mataric, "Cover me! a self-deployment algorithm for mobile sensor networks", *Proceedings of IEEE International Conference on Robotics and Automation*, ICRA'02, Washington, DC, pp. 80–91, May 2002.

[10] A. Howard, M. Mataric, G. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks", *Autonomous Robots Special Issue on Intelligent Embedded Systems*, pp.113–126, 2002.

[11] B. Yamauchi, "A frontier-based approach for autonomous exploration", *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, CIRA'97, Montere, CA, , pp. 146–156, June 1997.

[12] G. Wang, G. Cao, T. LaPorta, "A bidding protocol for deploying mobile sensors", *Proceedings of the 11th IEEE International Conference on Network Protocols*, ICNP'03, Atlanta, GA, pp. 80–91, November 2003.

[13] Datta, S., Klinowski, C., Rudafshani, M., Khaleque, S., "Distributed localization in static and mobile sensor networks", *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, p.p. 69 – 76, 19-21 June 2006.

[14] Li, X. and Santoro, N., "A ZONE-based Sensor Relocation Protocol for Mobile Sensor Networks", *Proceedings of 31st IEEE Conference on Local Computer Networks*, p.p. 923 – 930, Nov. 2006.

[15] Zhao, F. and Guibas, L., "Wireless sensor networks: an information processing approach", Boston: Elsevier-Morgan Kaufmann, 2004.

[16] Kwok, A. and Martinez, S., "Deployment algorithms for a power-constrained mobile sensor network", *IEEE International Conference on Robotics and Automation*, ICRA, p.p. 140 – 145, 19-23 May 2008.

DOI: 01.ijns.01.02.09